

.NET Framework

นายสาโรจน์ แซ่ลี้ ๕๔๑๐๕๑๓๐๖๙

นายสุวนนท์ ชูจันทร์ ๕๔๑๐๕๑๓๐๗๗

อาจารย์ที่ปรึกษา

ดร.อารีย์ ลิ้มวุฒิไกรจิรัฐ

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา สัมมนาทางการจัดการเทคโนโลยีสารสนเทศ  
คณะวิทยาการจัดการ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่

ภาคเรียนที่ ๑ ปีการศึกษา ๒๕๕๗

**คำนำ**

รายงานเล่มนี้จัดทำขึ้นเพื่อเป็นส่วนหนึ่งของวิชาสัมมนาทางการจัดการเทคโนโลยีสารสนเทศระดับชั้นปีการศึกษาที่ ๔ เพื่อให้ได้ศึกษาหาความรู้ในเรื่องของ .NET Framework เช่น คุณลักษณะ หลักการการทำงาน ความหมายของ CLR และ Class Library เป็นต้น และได้ศึกษาอย่างเข้าใจเพื่อเป็นประโยชน์กับการศึกษา

ผู้จัดทำหวังว่า  รายงานเล่มนี้จะเป็นประโยชน์กับผู้อ่าน หรือนักเรียน นักศึกษา ที่กำลัง หาข้อมูลเรื่อง .NET Framework หากมีข้อแนะนำหรือข้อผิดพลาดประการใด ผู้จัดทำขอน้อมรับไว้และขออภัยมา ณ ที่นี้ด้วย

คณะผู้จัดทำ

วันที่ ๑๑ พฤศจิกายน ๒๕๕๗

**สารบัญ**

**เรื่อง หน้า**

คำนำ ข

สารบัญ ค

สารบัญภาพ ง

สารบัญตาราง จ

บทนำ ๑

ภาพรวมของ .NET Framework ๓

ความต้องการระบบ ๘

การติดตั้ง .NET Framework 4.5, 4.5.1, และ 4.5.2 ๑๑

Common Type System ๑๒

The .NET Framework and Out-of-Band Releases ๑๕

Common Language Runtime (CLR) ๑๗

แอพพลิเคชั่นโดเมน ๑๙

รันไทม์โฮส ๒๐

บรรณานุกรม ๒๑

**สารบัญรูป**

**เรื่อง หน้า**

รูปที่ ๑: แสดงถึงความสัมพันธ์ของ CLR และ Class Library ของแอพพลิเคชั่น ๔

และระบบโดยภาพรวม

รูปที่ ๒: ตัวอย่างการใช้คลาส System.Console ใน .NET Framework ๗

Class Library ที่ใช้งานใน Console Application

รูปที่ ๓: แสดงตัวอย่างการใช้ Enumerations ๑๓

รูปที่ ๔: แสดงการติดตั้ง NuGet OOB ๑๖

**สารบัญตาราง**

**เรื่อง หน้า**

ตารางที่ ๑: แสดงถึงความสัมพันธ์ของ CLR และ Class Library ของแอพพลิเคชั่น ๔

ตารางที่ ๒: แสดงการสนับสนุนการทำงานของ .NET Frameworkใน Client OS ต่างๆ ๘

ตารางที่ ๓: แสดงการสนับสนุนการทำงานของ .NET Frameworkใน Server OS ต่างๆ ๙

ตารางที่ ๔: แสดงความต้องการของระบบที่ต้องการการติดตั้ง .NET Framework ๑๐

ตารางที่ ๕: แสดงเวอร์ชั่นของ Common Language Runtime ๑๘

เทียบกับเวอร์ชั่นของ .NET Framework

**บทนำ**

**.NET Framework คืออะไร?**

.NET Framework คือ Runtime Execution Environment ที่ช่วยในการจัดหาบริการ (Service) ที่หลากหลายเพื่อที่จะสามารถใช้งานแอพพลิเคชั่นต่างๆ ได้ ซึ่งประกอบด้วย 2 ส่วนประกอบหลักๆ คือ

1. Common Language Runtime (CLR) ที่จะควบคุมเกี่ยวกับหน่วยความจำและบริการต่างๆ ของระบบหรือแอพพลิเคชั่น
2. .NET Framework Class Library ขนาดใหญ่ ที่นักพัฒนาโปรแกรมสามารถเรียกใช้งานได้ผ่านแอพพลิเคชั่นของตนเอง

**บริการที่ .NET Framework เตรียมการเพื่อการใช้งานแอพพลิเคชั่นมีดังนี้**

* **การจัดการหน่วยความจำระบบ** โดยทั่วไปนักพัฒนาโปรแกรมมีหน้าที่ในการเขียนโค๊ดเพื่อจองหรือยกเว้นการใช้พื้นที่ในหน่วยความจำ เพื่อที่จะควบคุมวัตถุ (Object) ให้สามารถทำงานได้ แต่ใน .NET Framework จะมี CLR ช่วยทำหน้าที่ในส่วนนี้แทน
* **มีประเภทของระบบที่พบได้ทั่วไป** ประเภทของระบบที่ถูกกำหนดขึ้นโดย .NET Framework เป็นประเภทพื้นฐานที่สามารถพบเห็นได้ในทุกๆภาษาคอมพิวพ์เตอร์
* **Class Library ขนาดใหญ่** ที่นักพัฒนาโปรแกรมสามารถเรียกใช้และเข้าถึงได้ง่าย เพื่อใช้ในการควบคุมการทำงานของภาษาระดับต่ำทั่วไปโดยที่ไม่จำเป็นต้องเขียนโค๊ดเพื่อควบคุมด้วยตนเอง
* **ขอบเขตของการพัฒนา Framework และเทคโนโลยี** ใน .NET Framework จะมีการจำกัดขอบเขตในการพัฒนาแอพพลิเคชั่นเช่น ASP.NET จะใช้ในการพัฒนา เว็บแอพพลิเคชั่น ADO.NET ใช้ในการเข้าถึงฐานข้อมูลเป็นต้น
* **มีภาษาที่สามารถทำงานร่วมกับภาษาอื่นๆได้** ใน .NET Framework จะมีโค๊ดสื่อกลางที่เรียกว่า Common Intermediate Language (CIL) ซึ่งจะช่วยในการคอมไพล์โค๊ดในภาษาคอมพิวเตอร์ระดับสูงภาษาใดภาษาหนึ่ง ให้สามารถเข้าถึงหรือใช้งานในภาษาคอมพิวเตอร์ระดับสูงอื่นๆ ได้ ซึ่งจะช่วยให้นักพัฒนาโปรแกรมสามารถสร้างแอพพลิเคชั่นในภาษาคอมพิวเตอร์ระดับสูงที่ตนเองถนัดได้
* **ทำงานในรุ่นที่ต่างกันได้** แอพพลิเคชั่นที่ใช้ .NET Framework รุ่นล้าสมัยในการพัฒนา สามารถเรียกใช้งานใน .NET Framework รุ่นล่าสุดได้ โดยไม่ต้องเปลี่ยนแปลงโค๊ด
* **มีการทำงานที่ช่วยเหลือกัน** จะช่วยแก้ปัญหาในกรณีที่มี CLR หลากหลายรุ่นในคอมพิวเตอร์เครื่องเดียวกัน โดยที่สามารถทำให้ CLR รุ่นต่างๆ ทำงานร่วมกันได้และสามารถใช้งาน .NET Framework ได้ตรงรุ่นกับที่แอพพลิเคชั่นต้องการ
* **สามารถใช้งานได้หลายเพลตฟอร์ม** Class Library ของ .NET Framework ทำให้นักพัฒนาโปรแกรมสามารถสร้างชุดการทำงานที่สามารถทำงานได้หลายแพลตฟอร์ม เช่นWindow 7, Windows 8, Windows 8.1, Window Phone and Xbox 360

# ภาพรวมของ .NET Framework

เทคโนโลยี .NET Framework จะช่วยสนับสนุนในการสร้างและการทำงานของ แอพพลิเคชั่นและบริการเว็ป XML ซึ่ง .NET Framework ถูกออกแบบมาเพื่อจุดประสงค์ดังต่อไปนี้

* เพื่อให้การเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming หรือ OOP) มีความสอดคล้องกัน ไม่ว่าจะช่วยในด้านการจัดเก็บโค๊ดของวัตถุ การดำเนินงานในพื้นที่ส่วนตัว แต่สามารถใช้อินเตอร์เน็ตในการกระจายงานและดำเนินงานจากที่อื่นๆ และควบคุมจากระยะไกลได้
* เพื่อช่วยลดความต้องการของซอฟต์แวร์ที่ใช้ในการเปิดใช้งานแอพพลิเคชั่น
* เพื่อให้โค๊ดที่เขียนขึ้นมามีความปลอดภัย น่าเชื่อถือ
* ช่วยขจัดปัญหาในส่วนของประสิทธิภาพในการทำงาน
* สร้างระบบที่เป็นมาตรฐานโรงงานบนพื้นฐานของ .NET Framework ที่สามารถทำงานร่วมกับโค๊ดอื่นๆได้

.NET Framework ประกอบด้วย Common Language Runtime (CLR) ซึ่งเป็นพื้นฐานของ .NET Framework และ .NET Framework class Library คุณสามารถสร้างการทำงานในขณะที่งานหลักกำลังดำเนินการอยู่ เช่น การจัดการหน่วยความจำ การจัดการหน่วยการทำงานย่อย (Thread) และการควบคุมการทำงาน เป็นต้น ในขณะที่งานนั้นมีความปลอดภัยและความแม่นยำสูงในการทำงาน

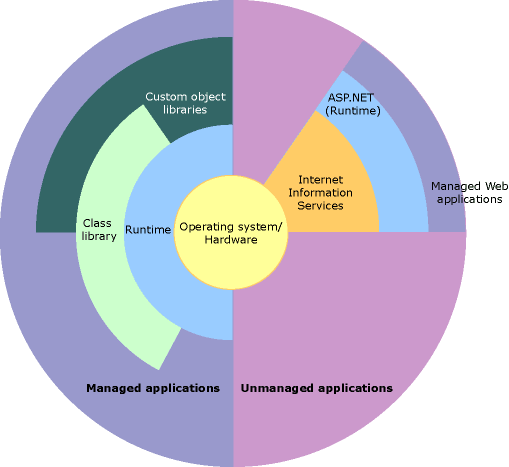
แนวคิดในการจัดการโค๊ดเป็นหลักการพื้นฐานของรันไทม์ ซึ่งโค๊ดที่มีการทำงานแบบรัน-ไทม์คือโค๊ดที่ผ่านการบริหารจัดการแล้ว ส่วน Class Library ก็คือแหล่งรวมของวัตถุ (Object) ที่สามารถนำมาใช้ใหม่ในการพัฒนาโปรแกรมตั้งแต่ในส่วนของรูปแบบการรับคำสั่งแบบโค๊ดดิบ รูปแบบหน้าอินเตอร์เฟสผู้ใช้งาน (GUI) จนถึงการประยุกต์ใช้กับงานที่อยู่บนพื้นฐานของนวัตกรรมใหม่ล่าสุดโดย ASP.NET เช่น Web Form และ XML Web service

.NET Framework สามารถทำหน้าที่เป็นผู้ควบคุมงาน (Host) โดยไม่จำเป็นต้องจัดการส่วนประกอบต่างๆ (Component) ซึ่งภาระงานตรงนี้จะถูกส่งไปที่กระบวนการทำงานของ CLR และ ทำให้เกิดการดำเนินงานในการจัดการโค๊ด ดังนั้นการสร้างแอพพลิเคชั่นสามารถทำได้ทั้งแบบที่มีคุณสมบัติการจัดการและไม่มีคุณสมบัติการจัดการ และ .NET Framework ไม่เพียงแต่ให้การควบคุมรันไทม์ ที่หลากหลาย แต่ยังสนับสนุนนักพัฒนาโปรแกรมในการใช้งานรันไทม์ของตัวของเขาเองด้วย

สำหรับตัวอย่าง ASP.NET ผู้ควบคุมรันไทม์ จัดเตรียมงานให้มีความยืดหยุ่น ช่วยในการจัดการโค๊ดของเซิฟเวอร์ ซึ่ง ASP.NET ทำงานโดยตรงกับรันไทม์ในการใช้งาน ASP.net Application และ XML Web service ซึ่งทั้งสองอย่างนี้ จะกล่าวถึงต่อไปในหัวข้อนี้

Internet Explorer คือตัวอย่างของแอพพลิเคชั่นที่ไม่มีการจัดการอย่างเป็นระบบ แต่ผู้ควบคุมรันไทม์จะช่วยฝังการทำงานในการจัดการส่วนประกอบต่างๆ หรือ การควบคุม Windows Form ในไฟล์เอกสาร HTML ด้วยวิธีการนี้จึงมีความเป็นไปได้ที่จะสามารถจัดการโค๊ดบนมือถือ แต่ต้องมีการปรับปรุงในการจัดการโค๊ดที่ไม่น่าเชื่อถือและการจัดเก็บไฟล์

ภาพประกอบต่อไปนี้ จะแสดงให้เห็นถึงความสัมพันธ์ของ CLR และ Class Library ของ แอพพลิเคชั่นของคุณ และระบบโดยรวม ซึ่งจะแสดงให้เห็นถึงกระบวนการการจัดการโค๊ดที่ทำงานอยู่ในสถาปัตยกรรมขนาดใหญ่



รูปที่ ๑: แสดงถึงความสัมพันธ์ของ CLR และ Class Library ของแอพพลิเคชั่น

และระบบโดยภาพรวม

**คุณลักษณะของ Common Language Runtime (CLR)**

CLR จะช่วยในการจัดการหน่วยความจำ การจัดการหน่วยการทำงานย่อย (Thread) การตรวจสอบความปลอดภัยของโค๊ด และการบริการด้านอื่นๆของระบบ คุณลักษณะเหล่านี้จะอยู่ภายในการจัดการโค๊ดซึ่งจะทำงานอยู่บน CLR

ในการรักษาความปลอดภัย องค์ประกอบต่างๆจะถูกจัดการและให้คะแนนความไว้วางใจในระดับที่ต่างกัน ขึ้นอยู่จำนวนปัจจัยต่างๆและที่มาของโค๊ด (เช่น อินเตอร์เน็ต ระบบเครือข่ายในองค์กร ระบบเครือข่ายภายใน เป็นต้น) นั่นหมายถึงการจัดการองค์ประกอบต่างๆอาจจะได้รับอนุญาตหรือไม่ได้รับอนุญาตให้ดำเนินงานในส่วนของการเข้าถึงในไฟล์ต่างๆ การเข้าถึงรีจิสทรี หรือฟังก์ชั่นอื่นๆที่สำคัญ แม้ว่าจะใช้ Application ตัวเดียวกันก็ตาม

รันไทม์จะอนุญาตให้โค๊ดทำงานเฉพาะส่วนที่ได้รับอนุญาตเท่านั้น เช่นตัวอย่าง ผู้ใช้สามารถไว้วางใจได้ว่าการทำงานที่ฝังอยู่บนหน้าเว็บไซต์สามารถเล่นภาพเคลื่อนไหวได้บนหน้าจอ แต่ไม่สามารถเข้าถึงข้อมูลส่วนตัว ไฟล์ระบบ หรือระบบเครือข่ายของผู้ใช้ได้

รันไทม์จะเป็นตัวช่วยที่ทำให้โค๊ดมีประสิทธิภาพด้วยการตรวจสอบโค๊ดและชนิดข้อมูลอย่างเข้มงวดให้ตรงตามโครงสร้างที่ออกแบบไว้ ซึ่งจะเรียกตัวช่วยตรงนี้ว่า Common Type System (CTS) ซึ่งจะทำหน้าที่ตรวจสอบให้แน่ใจว่าโค๊ดทั้งหมดนั้นมีความหมายในตนเอง ในหลายๆภาษาของ Microsoft หรือที่อื่นๆ จะสามารถคอมไพเลอร์ทำให้เกิดโค๊ดที่ถูกจัดการแล้วให้มีความสอดคล้องกับ CTS นั่นหมายถึงโค๊ดที่ถูกจัดการแล้วนั้นจะมีชนิดข้อมูลที่เชื่อถือได้และมีความปลอดภัย

นอกจากนี้สภาพแวดล้อมที่มีการจัดการของรันไทม์จะช่วยกำจัดหลายปัญหาซอฟต์แวร์ที่พบบ่อย ตัวอย่างเช่นรันไทม์ที่ทำงานจะควบคุมรูปแบบของวัตถุ (Object Layout) และจัดการการอ้างอิงถึงวัตถุอัตโนมัติ หลังจากนั้นจะปล่อยการควบคุมเมื่อวัตถุนั้นไม่ถูกนำมาใช้งาน

การจัดการหน่วยความจำอัตโนมัติจะช่วยแก้ปัญหา 2 ข้อผิดพลาดของ Application ที่พบบ่อยมากที่สุดคือการรั่วไหลของหน่วยความจำและการอ้างอิงหน่วยความจำที่ไม่ถูกต้อง

รันไทม์ช่วยลดระยะเวลาในการพัฒนาโปรแกรม ก็คือนักพัฒนาโปรแกรมสามารถสร้าง Application ด้วยภาษาที่ตนเองสามารถเลือกได้ โดยที่ยังได้รับประโยชน์จากรันไทม์ Class Library และส่วนประกอบต่างๆ ที่ถูกเขียนโดยนักพัฒนาโปรแกรมคนอื่นๆ

คุณลักษณะที่เรียกว่า just-in-time (JIT) จะช่วยคอมไพล์โค๊ดที่จะทำงานในภาษาเครื่องของระบบให้มีการทำงาน หมายความว่าการจัดการหน่วยความจำจะลบบางชิ้นส่วนของหน่วยความจำและเพิ่มพื้นที่ในการอ้างอิงหน่วยความจำเพื่อเพิ่มประสิทธิภาพในการทำงาน

**.NET Framework Class Library**

คือพื้นที่ที่เก็บรวบรวมชนิดของข้อมูลที่สามารถนำมาใช้ใหม่ได้ซึ่งทำงานร่วมกับ CLR ที่ซึ่งเป็นแหล่งของวัตถุ (Object) ที่จัดเตรียมไว้สำหรับนักพัฒนาโปรแกรมเรียกใช้ฟังก์ชั่นต่างๆ เพื่อพัฒนา Application ของตน Class Library ไม่เพียงแต่ช่วยให้การทำงานง่ายขึ้นเท่านั้น แต่จะช่วยลดระยะเวลาในการเรียนรู้คุณลักษณะที่ปรับปรุงใหม่ใน .NET Framework รุ่นใหม่ๆ ช่วยในการจัดการสตริง เก็บข้อมูล การเชื่อมต่อฐานข้อมูล การเข้าถึงไฟล์ และยังทำให้องค์ประกอบอื่นๆสามารถผสมผสานได้อย่างลงตัวกับคลาส (Classes) ใน .NET Framework

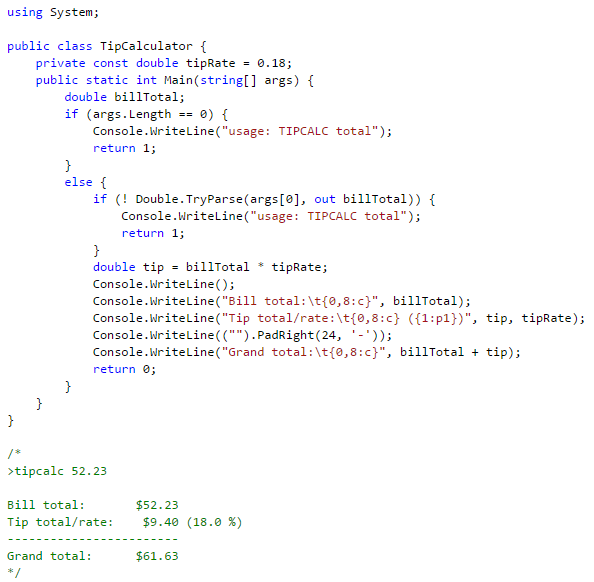
ตัวอย่างเช่น ชุดของคลาสใน .NET Framework เป็นเครื่องมือในการจัดการชุดของอินเตอร์เฟซ ซึ่งคุณสามารถใช้เพื่อการพัฒนาคลาสของตนเอง คลาสของคุณจะผสมผสานกันอย่างลงตัวกับคลาสใน .NET Framework และคุณสามารถใช้ .NET Framework ในการพัฒนา Application หรือบริการต่างๆได้ ดังนี้

* Console applications

แอพพลิเคชั่นใน .NET Framework สามารถใช้คลาส System.Console ในการอ่านตัวหนังสือและเขียนตัวหนังสือใน Console ข้อมูลจาก Console จะถูกอ่านจากสตรีมอินพุต ข้อมูลใน Console จะถูกเขียนไปที่สตรีมเอาต์พุต และข้อมูลที่ผิดพลาดจะถูกเขียนไปที่สตรีมเอาต์พุตของข้อผิดพลาด สตรีมจะเข้าไปทำงานอัตโนมัติเมื่อโปรแกรมเริ่มต้นและจะแสดงคุณสมบัติ ขาเข้า ขาออก และข้อผิดพลาด ตามลำดับ

ค่าของคุณสมบัติขาเข้าจะอยู่ในออบเจ็ค System.IO.TextReader ในขณะที่ค่าของคุณสมบัติขาออกและข้อผิดพลาดจะอยู่ในออบเจ็ค System.IO.TextWriter คุณสามารถเชื่อมโยงคุณสมบัติเหล่านี้ด้วยสตรีม ซึ่งไม่ได้เกี่ยวข้องกับ Console ทำให้สตรีมสามารถระบุพื้นที่ที่แตกต่างกันได้ในการป้อนข้อมูลหรือการส่งออกข้อมูล และคุณยังสามารถเปลี่ยนเส้นทางของการส่งออกข้อมูลไปยังไฟล์โดยการตั้งค่าคุณสมบัติขาออกใน System.IO.StreamWriter ซึ่งจะห่อหุ้ม FileStream ด้วยเม-ธอด Console.SetOut ซึ่งคุณสมบัติของ Console.In และ Console.Out ไม่จำเป็นต้องอ้างถึงสตรีมเดียวกัน

ตัวอย่างการใช้คลาส System.Console ใน .NET Framework Class Library ที่ใช้งานใน Console Application



คลาส System.Console

ผลจากการทำงาน Console Application

รูปที่ ๒: ตัวอย่างการใช้คลาส System.Console ใน .NET Framework

Class Library ที่ใช้งานใน Console Application

* Windows GUI applications (Windows Forms).

Windows Forms เป็นฟอร์มพื้นฐานของแอพพลิเคชั่น แบบฟอร์มจะเป็นหน้าว่างเปล่า ให้นักพัฒนาโปรแกรมได้สร้างอินเตอร์เฟสและเขียนโค๊ดเพื่อจัดการกับข้อมูลที่ทำงานร่วมกับ .NET Framework โดยที่ใช้ Visual Studio ในการควบคุม (เรียกโปรแกรมที่ใช้ในการจัดโค๊ดหรือสภาพแวดล้อมต่างๆว่า integrated development environment หรือ IDE) ซึ่งทำให้การทำงานเป็นไปอย่างง่ายและรวดเร็วในโซลูชั่นที่คุณต้องการ

* Windows Presentation Foundation (WPF) applications.
* ASP.NET applications.
* Windows services.
* Service-oriented applications using Windows Communication Foundation (WCF).
* Workflow-enabled applications using Windows Workflow Foundation (WF).

**ความต้องการระบบ**

**Hardware requirements**

|  |  |
| --- | --- |
| **Processor** | ๑ GHz |
| **RAM** | ๕๑๒ MB |
| **Disk** **space** **(minimum)** |  |
| **๓๒-bit** | ๘๕๐ MB |
| **๖๔-bit** | ๒ GB |

ตารางที่ ๑: แสดงความต้องการทางด้านฮาร์ดแวร์ในการติดตั้ง .NET Framework

**Supported client operating system**

|  |  |  |
| --- | --- | --- |
| **Operating system** | **Supported editions** | **Additional information** |
| Windows 8.1 | **๓๒**-bit and **๖๔**-bit | มีให้ใช้งาน .NET Framework ๔.๕.๑ อยู่แล้ว |
| Windows 8 | **๓๒**-bit and **๖๔**-bit | มีให้ใช้งาน .NET Framework ๔.๕ อยู่แล้ว |
| Windows 7 SP1 | **๓๒**-bit and **๖๔**-bit | - |
| Windows Vista SP2 | **๓๒**-bit and **๖๔**-bit | - |

ตารางที่ ๒: แสดงการสนับสนุนการทำงานของ .NET Frameworkใน Client OS ต่างๆ

**Supported server operating systems**

|  |  |  |
| --- | --- | --- |
| Operating system | Supported editions | Additional information |
| Windows Server 2012 R2 | **๖๔**-bit | มีให้ใช้งาน .NET Framework ๔.๕.๑ อยู่แล้ว |
| Windows Server 2012  (64-bit edition) | **๖๔**-bit | มีให้ใช้งาน .NET Framework ๔.๕ อยู่แล้ว |
| Windows Server 2008 R2 SP1 | **๖๔**-bit | .NET Framework สนับสนุนกับ Server Core Role SP1 หรือ ล่าสุด แต่ไม่สนับสนุนบน Windows Server 2008 R2 สำหรับ Itanium-Based Systems |
| Windows Server 2008 SP2 | **๓๒**-bit and **๖๔**-bit | .NET Framework ไม่สนับสนุนกับ Server Core Role. |

ตารางที่ ๓: แสดงการสนับสนุนการทำงานของ .NET Frameworkใน Server OS ต่างๆ

.NET เป็นส่วนหนึ่งในการแอพพลิเคชั่น บน Windows และยังมีการใช้ฟังชั่นรวมกันหลายๆตัวเพื่อทำการเปิดแอพพลิเคชั่นสำหรับการพัฒนา .NET Framework เป็นตัวช่วยที่ทำให้การเขียนโปรแกรมสามารถเขียนได้อย่างครอบคลุม และสอดคล้องกับโมเดลที่ใช้สร้างแอพพลิเคชั่น

สามารถดูตารางต่อไปนี้เพื่อดูความต้องการของระบบที่ต้องการ ก่อนที่จะติดตั้ง .NET Framework

|  |  |  |  |
| --- | --- | --- | --- |
| .NET Framework version | Developer installation | **Redistributable installation** | **Platform support** |
| ๔.๕.๒ | สามารถใช้งานกับ Visual Studio 2013, Visual Studio 2012 หรือ IDEs อื่นๆ | http://www.microsoft.com/en-us/download/details.aspx?id=42637  http://www.microsoft.com/en-us/download/details.aspx?id=42643  http://www.microsoft.com/en-us/download/details.aspx?id=42642 | สามารถติดตั้งลงบน Windows 8.1 และ windows ก่อนหน้านี้ อีกทั้งยังลงได้กับ Windows Server 2012 และ เวอร์ชั่นก่อนหน้านี้ |
| ๔.๕.๒ | สามารถใช้งานกับ Visual Studio 2012 หรือ IDEs อื่นๆ | http://www.microsoft.com/en-us/download/details.aspx?id=40772  http://www.microsoft.com/en-us/download/details.aspx?id=40773  http://www.microsoft.com/en-us/download/details.aspx?id=40779 | มีให้ใช้งานใน Windows 8.1, Windows Server 2012 R2 และ Visual Studio 2013  สามารถติดตั้งลงบน Windows 8 และ Windows ก่อนหน้านี้ อีกทั้งยังลงได้กับ Windows Server 2012 และ ก่อนหน้านี้ |
| ๔.๕ | ใช้งานร่วมกัน Visual Studio 2012 และ Windows SDK | http://www.microsoft.com/en-us/download/details.aspx?id=30653 | มีให้ใช้งานใน Windows 8, Windows Server 2012 และ Visual Studio 2012  สามารถติดตั้งลงบน Windows 7 และ Windows ก่อนหน้านี้ อีกทั้งยังลงได้กับ Windows Server 2008 SP2 และ ก่อนหน้านี้ |

ตารางที่ ๔: แสดงความต้องการของระบบที่ต้องการการติดตั้ง .NET Framework

**การติดตั้งแบบ .NET Framework Developer Pack**

.NET Framework Developer Pack เป็น .NET ที่ใช้กับนักพัฒนาโปรแกรม มักใช้กับ Visual Studio ซึ่งคุณสามารถเลือกโหลดได้ดังนี้

เวอร์ชั่น ๔.๕.๒ (http://www.microsoft.com/en-us/download/details.aspx?id=42637) สามารถติดตั้งลงบน Windows 8.1 และ windows ก่อนหน้านี้ อีกทั้งยังลงได้กับ Windows Server 2012 และ ก่อนหน้านี้ สามารถติดตั้งบน Visual Studio 2013, Visual Studio 2012 หรือ IDEs อื่นๆ

เวอร์ชั่น ๔.๕.๑ (http://www.microsoft.com/en-us/download/details.aspx?id=40772) สามารถติดตั้งบน Visual Studio 2012 หรือ IDEs อื่นๆ

**การติดตั้งและดาวน์โหลดแบบ .NET Framework redistributable**

* 1. สามารถดาวน์โหลด .NET Framework เวอร์ชั่นที่คุณต้องการ:

.NET Framework ๔.๕.๒

* Web Installer http://www.microsoft.com/en-us/download/details.aspx?id=42643
* Offline Installer http://www.microsoft.com/en-us/download/details.aspx?id=42642

.NET Framework ๔.๕.๑

* Web Installer http://www.microsoft.com/en-us/download/details.aspx?id=40773
* Offline Installer http://www.microsoft.com/en-us/download/details.aspx?id=40779

.NET Framework ๔.๕

* http://www.microsoft.com/en-us/download/details.aspx?id=30653

1. เลือกภาษาในเพจที่คุณทำการดาวน์โหลด
2. เลือกปุ่มดาวน์โหลด
3. สามารถเลือก Run หรือ Save เพื่อทำการดาวน์โหลด
4. ติดตั้ง .NET Framework

**Common Type System**

**ชนิดใน .NET Framework**

**คลาส (Classes)**

คลาสเป็นประเภทการอ้างอิงประเภทหนึ่ง ที่สามารถใช้ได้โดยตรงจากคลาสอื่น และได้รับมาจาก System.Object อีกด้วย คลาสจะทำงานในส่วนที่เป็น object ซึ่งสามารถดำเนินการและดึงข้อมูลใน Object Contains

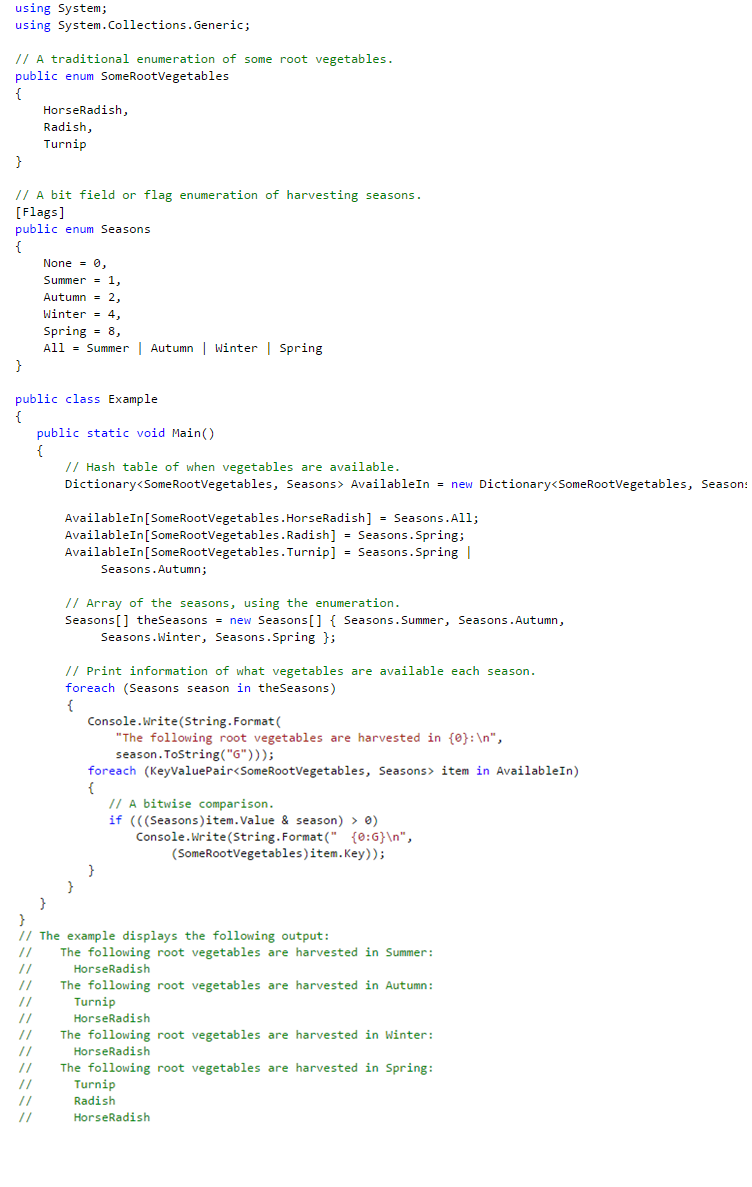
**โครงสร้าง (Structures)**

โครงสร้างจะมีประเภทของค่าที่เกิดขึ้นจาก System.ValueType โดยจะได้ค่ามาจาก System.Object ก่อน โครงสร้างนั่นมีประโยชน์มากมายสำหรับคำนวณค่าของหน่วยความจำ และค่าพารามิเตอร์ โดยข้อมูลจะถูกส่งจาก พารามิเตอร์ไปยังเมธอดที่มีพารามิเตอร์ที่ประเภทเดียวกัน

ใน Class Library ทุกประเภทของข้อมูล (Boolean, Byte, Char, DateTime, Decimal, Double, Int16, Int32, Int64, SByte, Single, UInt16, UInt32, and UInt64) จะถูกกำหนดให้เป็นข้อมูลแบบโครงสร้าง ในคราสที่มีความคล้ายคลึงกัน ข้อมูล และ การดำเนินงานที่เกี่ยวกับข้อมูลจะถูกกำหนดโครงสร้างในรูปแบบเมธอด ซึ่งหมายความว่าคุณสามารถเรียกเมธอดจากโครงสร้าง รวมถึงเมธอดที่มีมาจากระบบ หรือเรียกว่า Vrtual Methods ซึ่งจะถูกกำหนดไว้ที่ คลาส System.Object และ System.Value ในหมายความอื่นๆ โครงสร้างสามารถมีฟิลด์ คุณสมบัติ และเหตุการณ์ เช่นเดียวกับ เมธอดประเภท คงที่ และ ไม่คงที่ ซึ่งคุณสามารถสร้างโครงสร้างผ่านพารามิเตอร์

**Enumerations**

การนับ (เรียกสั้นๆว่า Enum) เป็นประเภทของข้อมูลที่ได้รับโดยตรงจาก System.Enum ประเภทการนับนั้นจะมีการตั้งชื่อ และชื่อที่ตั้งนั้นต้องไม่ซ้ำกับชื่ออื่น หรือไม่ลงชื่อแบบชนิดจำนวนเต็ม (Byte, Int32, or UInt64) โดยฟิลด์จะเป็นตัวอักษรคงที่ซึ่งจะแสดงความต่อเนื่อง ค่าบางค่าสามารถกำหนดฟิลด์ได้หลายแบบ เมื่อสิ่งนี้เกิดขึ้น คุณต้องทำเครื่องหมายให้กับค่านั่นสำหรับการส่งคืนและแปลงตัวอักษร คุณสามารถกำหนดค่าใดๆให้อยู่ในรูปแบบการนับ และในทางกลับกัน คุณสามารถสร้างตัวอย่างของการนับและการเรียกใช้เมธอดจาก System.Enum

****ตัวอย่าง Enum

รูปที่ ๓: แสดงตัวอย่างการใช้ Enumerations

**อินเตอร์เฟซ (Interface)**

สามารถระบุความสัมพันธ์ หรือ มีความสัมพันธ์อินเตอร์เฟซมักจะมีการใช้ฟังก์ชั่น เช่น การเปรียบเทียบและการเรียงลำดับ การทดสอบ หรือแจงรายการ และมีคุณสมบัติ เมธอด และเหตุการณ์ ซึ่งทั้งหมดนี้จะให้สมาชิกเป็นผู้ใช้งาน แม้ว่าอินเตอร์เฟซจะถูกกำหนดโดยผู้ใช้งาน มันก็สามารถแยกประเภทของการใช้งานของแต่ละผู้ใช้งานได้ ซึ่งหมายความว่าในแต่ละคลาสนั้นๆ จะมีโครงสร้างที่ทำงานติดต่อกับ อินเตอร์เฟซ

ข้อจำกัดเกี่ยวกับอินเตอร์เฟซ

* อินเตอร์เฟซสามารถแสดลงความสามารถต่างๆได้ แค่อินเตอร์เฟซสำหรับสมาชิก จะมีการเข้าถึงในแบบทั่วไป
* อินเตอร์เฟซไม่สามารถกำหนดโครงสร้างได้
* อินเตอร์เฟซไม่สามารถกำหนดฟิวต์
* อินเตอร์เฟซสามารถกำหนดได้เฉพาะในส่วนผู้ใช้งาน

**Delegates**

เป็นการอ้างอิงถึงสิ่งที่คล้ายคลึงกัน ในการทำงานใน C++ ซึ่งใช้สำหรับจัดการเหตุการณ์ และ ฟังก์ชั่น การเรียกกลับใน .NET Framework โดยจะมีความแตกต่างในการทำงาน มีการรักษาความปลอดภัย การตรวจสอบ และความปลอดภัย ซึ่งสามารถเป็นตัวแทนของเมธอด หรือ เมธอดคงที่เข้ากันได้

พารามิเตอร์ สำหรับ Delegates จะมีการทำงานกับพารามิเตอร์ที่สอดคล้องกัน แต่มีข้อจำกัดอยู่บ้างในการทำงาน เพราะ ต้องมีรับรองว่าการส่งอาร์กิวเมนต์สามารถส่งผ่านไปได้อย่างปลอดภัย ในทำนองเดียวกัน delegate มีการส่งค่ากลับไปยังเมธอด การส่งค่ากลับจะมีความเข้มงวดมากกว่าการส่งค่าแบบอื่น เพราะต้องมีการรับรองว่าสามารถส่งข้อมูลได้อย่างปลอดภัย

Delegates มีการสืบทอดจาก System.MulticastDelegate ซึ่งสืบทอดมาจาก System.Delegate

ในส่วนของภาษา C#, Visual Basic, และ C++ ไม่อนุญาติให้มีการถ่ายทอด

**The .NET Framework and Out-of-Band Releases**

.NET Framework มีการพัฒนาให้มีการรองรับหลายๆแพลตฟอร์ม เช่น Windows Phone และ Windows Store Apps อีกทั้งยังมี App Desktop แบบดั้งเดิม และ Web App อีกด้วย ในส่วนนี้เราจะอธิบายถึง OOB (Out-of-Band)และคุณลักษณะใหม่ๆของ .NET Framework

**ประโยชน์ของ OOB**

ในการส่งสินค้า หรือ การปรับปรุงส่วนประกอบนั้น Microsoft จะให้ OOB ในการอัพเดท .NET Framework อีกทั้งยังส่วนในการรวบรวมและตอบสนองต่อความคิดเห็นของลูกค้าได้อย่างรวดเร็ว

เมื่อใช้ OOB ในแอพพลิเคชั่นของคุณ คุณไม่จำเป็นต้องติดตั้ง .NET Framework เวอร์ชั่นล่าสุดเลย เพราะ OOB จะมีการตรวจสอบและทำการปรับปรุงเวอร์ชั่นอัตโนมัติ

**แพ็คเกจ OOB มีการจัดการอย่างไร**

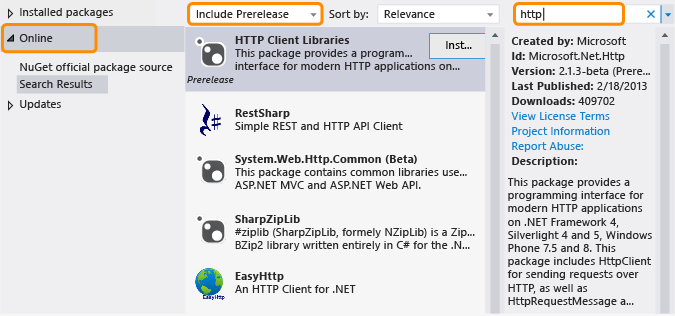
OOB ถูกออกแบบมาให้ใช้งานกับ CLR เป็นหลัก ซึ่งจะถูกจัดการผ่าน NuGet Package Manager โดย NuGet จะช่วยให้คุณสามารถเรียกดูและเพิ่ม Libraries ได้อย่างง่ายดาย ในทุกเวอร์ชั่นของโปรแกรม Visual Studio จะมี NuGet อยู่แล้ว หากไม่ได้ติดตั้งคุณสามารถไปยังเมนู Tool เพื่อทำการติดตั้งได้

**การใช้งานแพ็คเกจ NuGet OOB**

เมื่อคุณติดตั้ง NuGet แล้ว คุณสามารถค้นหา และ เพิ่ม ส่วนเสริมต่างๆที่เกี่ยวข้องกับ NuGet  
๑. เปิด Visual Studio และ เลือก เมนู Manage NuGet packages

๒. เลือกแถบ Online

๓. ให้เลือก Include Prerelease คุณสามารถหาส่วนเสริมที่คุณต้องการได้ในกล่องตรงกลาง   
๔. ในส่วนด้านขวาคุณสามารถใช้ค้นหาสิ่งที่คุณต้องการ

****

รูปที่ ๔: แสดงการติดตั้ง NuGet OOB

**Common Language Runtime (CLR)**

.NET Framework จะมีการเรียกใช้ Common Language Runtime เพื่อทำการเปิด code และ บริการที่ทำให้กระบวนการพัฒนาได้ง่ายขึ้น

คอมไพเลอร์ และ เครื่องมือที่ใช้แสดงฟังก์ชั่น Common Language Runtime และช่วยให้คุณสามารถเขียนโค๊ดได้อย่างมีประสิทธิภาพ สามารถดึงทรัพยากรออกมาให้มีประโยชน์ที่สุด

โค๊ดที่คุณพัฒนาขึ้น จะมีการเรียกคอมไพล์เลอร์เพื่อจัดการโค๊ด ซึ่งมีประโยชน์ทั้งการ ทำงานข้ามภาษา ความปลอดภัย เวอร์ชั่นที่ทันสมัยและการสนับสนุนการพัฒนา รูปแบบที่เรียบง่ายสำหรับความสัมพันธ์ของส่วนประกอบ แก้จุดบกพร่่อง และ บริการ Profiling

เมื่อต้องการใช้งานบริการการจัดการโค๊ด Language Compilers จะต้องแสดง Metadata ที่อธิบายถึงประเภท สมาชิก และการอ้างอิงในโค๊ดของคุณ Metadata จะถูกเก็บไว้กับโค๊ดสามารถใช้ได้ทุกเวอร์ชั่นของ CLR แบบพกพา (PE)

รันไทม์จะใช้ Metadata เพื่อค้นหาและโหลดคลาสเลเอาท์ของหน่วยความจำ แก้ไขคำร้องเมธอดการสร้างรหัส ความปลอดภัย และการกำหนดขอบเขตเวลาการทำงาน และจะมีการจัดการออบเจ็คแบบอัตโนมัติและจัดการการอ้างอิงไปยังออบเจ็ค และการยกเลิกเมื่อไม่ได้ใช้งาน

ออบเจ็คที่มีอายุการใช้งานจะมีการจัดการด้วยวิธีที่เรียกว่า การจัดการข้อมูล มีการเก็บไฟล์ขยะของหน่วยความจำรวมถึงข้อผิดพลาดของโปรแกรมอื่นๆ หากโค๊ดของคุณถูกจัดการ คุณสามารถใช้การจัดการใน .NET Framework ได้เลย

CLR ได้ถูกออกแบบมาให้สามารถใช้งานได้ง่าย และ การโต้ตอบระหว่างออบเจ็คข้ามภาษา ออบ-เจ็คที่เขียนต่างภาษากันสามารถสื่อสารกันได้ ซึ่งการทำงานข้ามภาษานั้นสามารถเป็นไปได้ เพราะ คอมไพล์เลอร์และเครื่องมือกำหนดให้รันไทม์สามารถทำงานโดยใช้ Common Type System Defined และปฏิบัติตามกฏของรันไทม์ สำหรับการสร้างประเภทขึ้นมาใหม่ เช่นเดียวกับการสร้าง ใช้งาน การทำให้คงอยู่ และความสัมพันธ์ของประเภท และเป็นส่วนหนึ่งของ Metadata ซึ่งทุกๆองค์ประกอบที่จัดการข้อมูลสารสนเทศเกี่ยวกับส่วนประกอบและทรัพยากรที่ถูกสร้างขึ้นมา

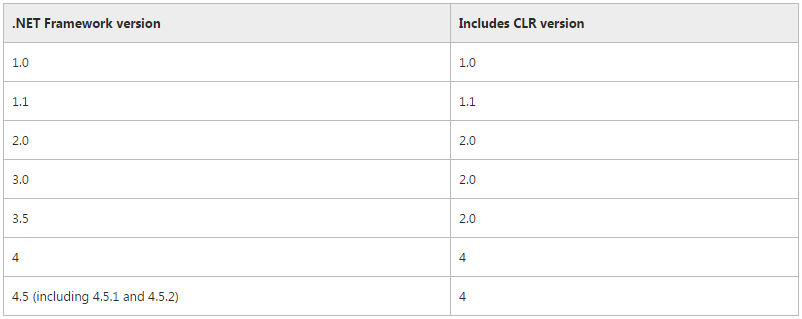
รันไทม์ใช้ข้อมูลสารสนเทศเพื่อให้แน่ใจว่าองค์ประกอบ หรือ แอพพลิเคชั่นนั่นมีเวอร์ชั่นที่ตรงตามต้องการของระบบ ซึ่งจะช่วยลดข้อผิดพลาดในการทำงานลง ข้อมูลการลงทะเบียนและข้อมูลสถิติ จะไม่ถูกจัดเก็บลงไปในรีจิสทรี เพราะการสร้างที่ยุ่งยาก และการรักษาความปลอดภัย แต่ข้อมูลประเภทที่เกี่ยวข้องจะถูกกำหนดให้ถูกเก็บไว้ในรูปแบบโค๊ดใน Metadata ทำให้การจำลององค์ประกอบ และการลบ ให้มีความซับซ้อนน้อยลง

ภาษาคอมไพเลอร์และเครื่องมือ จะแสดงการทำงานของรันไทม์ให้อยู่ในรูปแบบของฟังก์ชั่น เพื่อวัตถุประสงค์ให้เกิดประโยชน์สูงสุด และง่ายต่อการพัฒนา

ประโยชน์ของรันไทม์มีดังนี้

* ประสิทธิภาพการทำงาน
* ความสามารถในการใช้ส่วนประกอบที่มีการพัฒนาในภาษาอื่นๆ
* การบริการประเภท จาก Class Library
* คุณสมบัติของภาษา เช่น อินเตอร์เฟซ และการทำงานเกินขีดจำกัดสำหรับการเขียนโปรแกรมเชิงวัตถุ
* สนับสนุนโครงสร้างที่มีข้อจำกัด
* สนับสนุนแอตทริบิวต์
* การรวบรวมไฟล์ขยะ

**เวอร์ชั่นของ Common Language Runtime**

****

ตารางที่ ๕: แสดงเวอร์ชั่นของ Common Language Runtime

เทียบกับเวอร์ชั่นของ .NET Framework

**Application Domains**

Operating Systems และ Runtime Environments จะมีรูปแบบในการใช้งานบางส่วนที่แยกกันอยู่ เช่น Windows จะแยกกระบวนการเพิ่อแน่ใจว่ารหัสที่ใช้ในโปรแกรมจะไม่ส่งผลกระทบกับส่วนอื่นๆ Application domains จะมีการแยกสัดส่วนของความปลอดภัย ความน่าเชื่อถือ และเวอร์ชั่น แอพพริเคชั่นโดเมนถูกสร้างขึ้นจากรันไทม์โฮสซึ่งเป็นผู้รับผิดชอบในการบูตสเตป กับ common language runtime ก่อนทำการเปิดโปรแกรม

**Application domains and Threads**

หน่วยการทำงานย่อย (Thread) หลายๆตัวสามารถทำงานใน Single Application Domain ได้ตลอดเวลา โดยเฉพาะอย่างยิ่งหน่วยการทำงานย่อยจะไม่ถูกจำกัดว่าต้องอยู่กับโปรแกรมแค่ตัวเดียว ซึ่งหน่วยการทำงานย่อยจะมีอิสระในการเปลี่ยนโยกย้ายการทำงานได้ ในเวลาใดๆก็ตามจะมีเทรดทำงานอยู่ใน Application domains ไม่ว่าจะเป็นหน่วยการทำงานย่อย 0 1 หรือแบบหลายๆตัวจะมีการทำงานกับ Application domains ที่ได้กำหนดไว้ รันไทม์จะติดตามการทำงานของหน่วยการทำงานย่อยที่กำลังทำงานกับ Application domains ซึ่งคุณสามารถค้นหาโดเมนที่ได้โดยการเรียกใช้เมธอด Thread.GetDomain

**Runtime Host**

ASP.NET ใช้ทำการโหลดรันไทม์ในการประมวลผลควบคุมจัดการคำข้อร้องที่มาถึงเว็บ ASP.NET ยังสร้างโดเมนสำหรับใช้งานเกี่ยวกับเว็บแอพพลิเคชั่นบนเว็บเซิฟเวอร์

Microsoft Internet Explorer ใช้สร้างแอพพริเคชั่นโดเมนในการดำเนินตัวควบคุมจัดการ โดย .NET Framework สนับสนุนการดาวน์โหลด และควบคุมการดำเนินงานบนโปรแกรมเว็บ รันไทม์อินเตอร์เฟซเป็นส่วนขยายจาก Microsoft Internet Explorer ทำการคัดลอกเพื่อสร้างแอพพลิเคชั่นโดเมนที่ใช้ในการดำเนินการควบคุมและจัดการ

Shell Executable จะมีการเรียกรันไทม์เพื่อควบคุมการถ่ายโอนข้อมูลไปยังโฮสติ้ง

**บรรณานุกรม**

ชื่อเรื่อง Getting Started with the .NET Framework. สืบค้นเมื่อวันที่ ๒๕ ตุลาคม, ปี พ.ศ.๒๕๕๗,

จากเว็บไซต์: http://msdn.microsoft.com/en-us/library/hh425099(v=vs.110).aspx